Using psycho-physiological measures to assess task difficulty in software development

Fritz, Thomas ; Begel, Andrew ; Müller, Sebastian C ; Yigit-Elliott, Serap ; Züger, Manuela

Abstract: Software developers make programming mistakes that cause serious bugs for their customers. Existing work to detect problematic software focuses mainly on post hoc identification of correlations between bug fixes and code. We propose a new approach to address this problem — detect when software developers are experiencing difficulty while they work on their programming tasks, and stop them before they can introduce bugs into the code. In this paper, we investigate a novel approach to classify the difficulty of code comprehension tasks using data from psycho-physiological sensors. We present the results of a study we conducted with 15 professional programmers to see how well an eye-tracker, an electrodermal activity sensor, and an electroencephalography sensor could be used to predict whether developers would find a task to be difficult. We can predict nominal task difficulty (easy/difficult) for a new developer with 64.99% precision and 64.58% recall, and for a new task with 84.38% precision and 69.79% recall. We can improve the Naive Bayes classifier's performance if we trained it on just the eye-tracking data over the entire dataset, or by using a sliding window data collection schema with a 55 second time window. Our work brings the community closer to a viable and reliable measure of task difficulty that could power the next generation of programming support tools.

DOI: https://doi.org/10.1145/2568225.2568266

Using Psycho-Physiological Measures to Assess Task Difficulty in Software Development

Thomas Fritz[†], Andrew Begel[⋆], Sebastian C. Müller[†], Serap Yigit-Elliott[◦], Manuela Züger[†]

[†]University of Zurich
Zurich, Switzerland

[⋆]Microsoft Research
Redmond, WA USA

[◦]Exponent
Bellevue, WA USA

ABSTRACT

Software developers make programming mistakes that cause serious bugs for their customers. Existing work to detect problematic software focuses mainly on *post hoc* identification of correlations between bug fixes and code. We propose a new approach to address this problem — detect when software developers are experiencing difficulty while they work on their programming tasks, and stop them before they can introduce bugs into the code.

In this paper, we investigate a novel approach to classify the difficulty of code comprehension tasks using data from psycho-physiological sensors. We present the results of a study we conducted with 15 professional programmers to see how well an eye-tracker, an electrodermal activity sensor, and an electroencephalography sensor could be used to predict whether developers would find a task to be difficult. We can predict nominal task difficulty (easy/difficult) for a new developer with 64.99% precision and 64.58% recall, and for a new task with 84.38% precision and 69.79% recall. We can improve the Naive Bayes classifier's performance if we trained it on just the eye-tracking data over the entire dataset, or by using a sliding window data collection schema with a 55 second time window. Our work brings the community closer to a viable and reliable measure of task difficulty that could power the next generation of programming support tools.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors, software psychology*

General Terms

Human Factors, Experimentation, Measurement

Keywords

psycho-physiological, task difficulty, study

ICSE '14, May 31 – June 7, 2014, Hyderabad, India

1. INTRODUCTION

Knowing how hard a task is as it is being performed can help in many dimensions. For instance, the estimate for completing a task might be revised or the likelihood of a bug occurring in the source code changes for the task might be predicted. Existing work to determine task difficulty has mainly focused on already existing artifacts, such as task descriptions, and the similarity of artifacts using machine learning classifiers. In our research, we are investigating a novel approach to determine task difficulty that uses psycho-physiological data gathered from the developer while he is working, such as electroencephalographic (EEG) activity along the forehead or electrodermal activity (EDA). By using psycho-physiological sensors and collecting data while a developer is performing a task, we present the first approach that can support an instantaneous measure of task difficulty that does not rely on already produced artifacts or even whether the developer is writing any code at all.

There has been extensive research in psycho-physiology investigating how various measures can be linked to psychological states and processes (e.g., [41, 59]), but only little work has investigated the use of such measures in software development. Predominantly this work used eye-tracking technology to retrospectively determine the effect on visual effort for different representations, such as differences in identifier styles or the visual representation of requirements (e.g., [66, 64]). None of this work has used psycho-physiological features of software developers to measure task difficulty. One preliminary pilot study by Parnin [54] has explored the use of electromyography to measure sub-vocal utterances and investigate them as an additional measure for task difficulty. While Parnin found a correlation between utterances and a developer editing code, his work looks at only a single psycho-physiological feature, ignoring the potential to look for instantaneous, or more general measures of psycho-physiological features corresponding to task difficulty.

In this paper, we investigate whether we can use psycho-physiological measurements to determine whether a code comprehension task is perceived as easy or difficult. In particular, we ask the following questions:

(RQ1) Can we acquire psycho-physiological measures from eye-tracking, EDA and EEG sensors to accurately predict whether a task is difficult or easy?

(RQ2) Which combination of psycho-physiological sensors and associated features best predict task difficulty?

(RQ3) Can we use psycho-physiological measures to predict task difficulty as the developer is working?

With such a code- and quality-independent indicator for a developer's difficulty with a task, it may be possible to design a set of interventions that could prevent the developer from introducing bugs caused by cognitive difficulties, and also provide timely support for the remainder of his task.

To answer our research questions, we conducted an exploratory study in which 15 professional software developers monitored with psycho-physiological sensors performed six to eight tasks. We gave the developers code comprehension tasks that were small, but large enough to challenge the subjects for a few minutes at a time. Using all of the sensor data, we were able to train a classifier to predict whether a developer, on which the classifier was not trained on, would perceive the task to be easy or difficult with 64.99% precision and 64.58% recall. Using just the eye-tracking data resulted in even greater predictive power. To create a classifier that can operate while the developer does his work, we explored how well a sliding time window data collection approach (adjusting the size of the time window from 5 second to 60 seconds, sliding it 5 seconds each time) could make predictions of the developer's final assessment of task difficulty after finishing his task. We found that combining subsets of sensors with particular time windows could improve classifier performance when predicting a new developer's task difficulty and a developer-task task difficulty pair.

Our contributions are

- an exploratory study on the viability of using psycho-physiological sensors to determine code comprehension task difficulty;
- an approach to classify tasks by difficulty using time intervals suitable for on-the-fly classification;
- and an investigation of which combination of psycho-physiological sensors and measurements are most effective at predicting task difficulty.

Overall, our work provides the software engineering research field with a new perspective on using psycho-physiological measures to understand and support the software developer in his activities. In the future, advances in sensor technology and data analysis techniques should make it possible to employ simpler, cheaper, and more accurate metrics and develop them into programming support tools.

2. RELATED WORK

Related work can be categorized into two areas: the general use of psycho-physiological measures to study psychological states and processes and research related to aspects of software development using psycho-physiological measures.

2.1 Using Psycho-Physiological Measures

There is a broad range of psycho-physiological measures that have been explored and linked to psychological, and specifically cognitive, processes and states. All of these measures have different strengths and weaknesses with respect to aspects such as invasiveness, sensitivity, generalizability, interpretability and ease of collecting [41, 59]. Some of the most commonly used measures can roughly be categorized into eye-related, brain-related or skin-related measures.

Eye. There is a variety of eye-related measures, such as the pupil size, fixation duration or number of saccades. Early on, Beatty found that task-evoked pupillary response, in particular the peak amplitude of the pupil diameter, is an indicator for memory load or also processing load and that it varies with task difficulty [6]. Further research on pupil size found similar correlations, e.g., to mental workload of subtasks [36] and cognitive load [40], and even used pupil dilation as a measure for workload at task boundaries [4]. Others used measures of fixation and saccades, e.g., Goldberg *et al.* found that a higher number of saccades is an indicator for a poorer interface [24], and in their overview on eye-tracking research in HCI and usability, Jacob *et al.* state that the mean fixation duration is believed to be an indicator of a participant's difficulty in extracting information from a display [37]. Recent approaches have also used eye-related measures to train machine learning classifiers and predict a person's cognitive state (e.g., [69, 21]).

While measures on pupil size, fixations and saccades are commonly captured using an eye-tracking sensor, eye-blink rate is better measured through electrodes placed around the eye, *i.e.* Electrooculography (EOG) or by filtering certain frequencies within Electroencephalography (EEG). Studies on eye-blink rate have shown that it is inversely correlated with attention or mental load, *i.e.* the lower the blink rate, the higher the mental load or attention (e.g., [72, 33, 34, 5]).

Brain. With brain-related measures we refer to the recording of electrical activity inside the brain or close to the surface of the scalp, *i.e.* Electroencephalography (EEG). Studies have shown that specific frequency bands, often referred to as alpha, beta, gamma, delta and theta, within the EEG data can be connected to different mental states [11]. For instance, several studies found that a decrease of alpha EEG activity and often an increase in theta EEG activity was accompanied with an increase in attentional demand and working memory load (e.g., [71, 23, 70]). Other studies examined an EEG task engagement index defined as "beta / (alpha + theta)"(e.g., [42, 43, 12]) based on evidence that with increases in task engagement, theta is suppressed, alpha is blocked and beta increases in relative power, or they found that the theta and delta band are sensitive to task difficulty manipulations (e.g., [15]). As with eye-tracking measures, researchers have also investigated using EEG data and machine learning to predict aspects, such as the working memory load or the cognitive task (e.g., [44, 26]).

Skin. Electrodermal activity (EDA), also known as skin conductance (SC) or galvanic skin response (GSR), has been closely linked with arousal, attention, emotional states, stress and anxiety [13, 17]. Frequently, features of electrodermal activity have been used in combination with measures such as blood-volume pressure and respiration to classify the data into classes or states of emotion (e.g., [55, 45]). In addition, studies have shown that EDA measures can be used to indicate cognitive load levels, task difficulty level and distinguish cognitive load at the workplace from stress (e.g., [68, 53, 62]). For instance, Nourbakhsh *et al.* have shown that normalized frequency domains of electrodermal activity were significant to indicate task difficulty levels for arithmetic and reading tasks [53]. Researchers have also investigated EDA as a real time measure, e.g., to adapt the workload of an operator and avoid it to become too high [29] or to detect emotions and improve the gaming experience [49].

Finally, researchers have combined various of these measures. Wilson, for instance, measured brain activity, eye blinks, electrodermal activity and heart rate and found that electrodermal activity measures as well as alpha and delta bands of brain activity showed significant changes to varying mental workload demands in flying scenarios, while the heart rate was less sensitive [74]. Similarly, Ryu *et al.* combined

multiple sensors and found that a combination worked well for distinguishing between the difficulty levels of tasks [60]. More recently, Haapalainen *et al.* collected data using multiple sensors, including a NeuroSky mindset for EEG, eye-tracking and a GSR armband and compared their ability to assess cognitive load using six elementary cognitive tasks with varying difficulty levels each. Their results show that electrocardiogram median absolute deviation and median heat flux measurements were most accurate to classify between low and high cognitive load [27].

Similar to the aforementioned research, we also want to take advantage of the psycho-physiological measures and differentiate between difficult and easy tasks. In particular, we are looking into a combination of sensors, similar to the ones by Haapalainen [27]. However, we are looking at aspects of software development and thus differ in the tasks and participants we are studying. In particular, the tasks in our study are more closely related to software development tasks and the participants are professional software developers. Since early research on reading algorithms has found differences to reading prose [16], our study provides insights on how these psycho-physiological measures could be used in the software development domain.

2.2 Psycho-Physiology in SD

A few studies have investigated the use of psycho-physiological measures in software development, mainly using eye-tracking. An early study on code comprehension by Crosby *et al.* used eye-tracking to study the scan patterns and strategies of high and low experience developers. In their study, they used eye fixation as a measure of attention, classified code into 5 categories from easy to hard and found that high-experienced developers use less time on comments and more time on complex statements than low experienced developers [16]. More recent studies by Bednarik *et al.* also analyzed differences in strategies for less and more experienced developers in program comprehension and debugging [8, 9].

Using eye-tracking technology, researchers have also studied the effect of different representations in software development on the visual effort. For instance, Sharif *et al.* looked at the effect of identifier naming conventions—camelCase and under_score—in code comprehension and found that the accuracy in answers stays the same but time and visual effort decreases [66]. While Sharafi *et al.* also looked at memorability of identifier styles they examined the impact of gender on source code reading and found different comprehension strategies in male and female subjects using eye-tracking [63]. Studies have also looked at differences in other representations, such as graphical and textual representations of code variables [52], requirements [64] as well as the representation and layout of design patterns [67, 57].

To study the link between code reviews and defect detection, researchers have also examined the time developers spend scanning code by summarizing fixation durations over specific areas of interest and counting the number of fixations. Thereby, they found that a longer scan time correlates significantly with a better defect detection [73, 65].

All of these approaches examine software engineering aspects, however, they are limited to eye-related features. Khan *et al.* have looked into another psycho-physiological aspect and its link to performance by investigating how the mood of developers affects debugging and programming [38, 39]. In these studies, different moods were induced by showing developers video clips and then the developers' performance was measured. None of this research has investigated the use of psycho-physiological measures for determining task difficulty. Closest to our work is a preliminary pilot study by Parnin, who has explored the use of electromyography to measure sub-vocal utterances [54]. From early results, he found that such a measure might be used to determine the difficulty of a programming task. While these initial results indicate the potential of these measures, this paper goes further in investigating multiple psycho-physiological measures and their relation to task difficulty in a study with fifteen professional software developers.

3. EXPERIMENT

We conducted a lab experiment with 15 professional software developers. Each subject performed eight code comprehension tasks as we recorded various psycho-physiological and subjective measurements[1].

3.1 Subjects

Subjects were recruited from a pool of professional software developers who lived in the greater Seattle, WA area and had registered their interest in participating in user studies at Microsoft. A screening questionnaire selected 20 of these candidates who had at least 2 years of software development experience, knew how to program in C# (and had done so in the last year), did not need to wear bifocal or trifocal glasses (they interfere with the eye-tracker), and were available to come to our lab. Five of the 20 selected subjects did not show up. Those that completed the 1.5 hour experiment were remunerated with a single license for their choice of Microsoft consumer software (a standard payment for Microsoft user study participants). Fourteen of the subjects were male and one was female. Subjects ranged in age from 27 to 60 years of age (mean 41.6, stdev 8.2).

3.2 Data Capture

We recorded study data using three psycho-physiological sensors: eye-tracking, EDA, and EEG. We also recorded the subject's think aloud narrative, recorded a video of the experiment, and recorded a screen capture. The subjects filled out a pre-questionnaire, a written NASA TLX survey [32] after completing each experimental task, and a post-questionnaire after the entire experiment that asked them to rank each of the tasks by relative perceived difficulty. The experiment administrator also took hand-written notes.

Eye-tracking has been used to assess task difficulty and cognitive and mental load [40, 36, 1, 15]. We used a Tobii TX300 eye-tracker using a 300 Hz tracking frequency to collect gaze location information, fixation and saccade count and duration and pupil diameter. The eye-tracker has an accuracy of 0.4° of visual angle, which is equivalent to 13 pixels on its built-in 96 dpi 1920 x 1080 23-inch monitor. We applied Tobii Studio's built-in I-VT fixation filter with default parameters in order to classify eye movements based on the velocity of shifts in the eyes' directions. To avoid gaze inaccuracy, we directed the tool and our subsequent analysis to record and analyze data only from the subject's dominant eye (determined as part of our experimental procedure).

[1]A replication package of the experiment is available via http://research.microsoft.com/apps/pubs/?id=209878

Electrodermal activity (EDA) is an oft-used sensor to detect arousal, particularly cognitively-determined arousal [13]. To measure EDA, we used an Affectiva Q Sensor 2.0 [56] worn on the wrist of the subject's non-dominant (and non-mouse-holding) hand. The Q Sensor samples at a rate of 8 Hz, simultaneously measuring skin temperature along with three-axis acceleration data. Data is stored on the device itself, and streamed via Bluetooth to a recording computer.

Electroencephalography (EEG) refers to the measurement of the brain's electrical activity that arises from neuronal firing [2]. It is used in a variety of fields, such as neurology and Brain-Computer Interface (BCI) research. There are a variety of devices that record multi-channel EEG signals using sensors attached with gel to various points on a subject's head. To make our experiment less invasive (and minimize cleanup), we decided to use an off-the-shelf NeuroSky MindBand EEG sensor. It is a one-channel, noise-canceling, dry sensor that records the EEG signal at 512 Hz from a single location on the subject's forehead (reading signals mainly from the pre-frontal cortex). The MindBand produces a single, pre-filtered, time-varying voltage signal, as well as two computed signals, Attention and Meditation, corresponding to paying attention and feeling calm and centered [51]. While these are both produced by proprietary (read: trade secret) algorithms, the signals are always available from the entire family of NeuroSky sensors.

Audio/video capture of the experiment was done with two 60 fps cameras, one pointed straight at the subject from the screen (like a webcam) and the other above the subject aimed at the screen and keyboard. The Tobii Studio v3.0 software that was used to run the eye-tracker also recorded the full resolution screen at 60 Hz and added a "follow-the-bouncing-ball" visualization on top of the recording to visualize the subject's gaze location.

We attempted to use all of the psycho-physiological sensors to record psycho-physiological data for all of our tasks. After refining our procedures with a two person pre-pilot, we were able to successfully capture the complete set of sensor signals for 12 of our 15 participants. We got eye-tracking data for everyone, EEG data for 13 out of 15 participants, and EDA data for 12 out of 15 participants. Two participants failed to produce a valid EEG signal; coincidentally, they failed to produce a measurable EDA signal as well. Another participant's EEG data was lost during capture.

3.3 Experimental Tasks

Subjects were asked to perform short (several minutes) code comprehension tasks. In two pre-pilots, we had asked subjects to perform more complex 15-30 minute tasks involving code comprehension and mental code execution, but we found it difficult to scale our characterization of our subject's activities to a granularity of tens of milliseconds for such a long period of time. We eventually designed smaller, shorter tasks, though still limited to code comprehension and mental execution. While these are much simpler than the tasks of many software professionals, we believe that this starting point helps us identify the big picture answers to our research questions, and leaves more details to future experiments.

Each subject was asked to work on ten tasks: two practice questions and eight which were measured. During each task, they were asked to read a short passage of C# code presented on a single screen in the Visual Studio 2012 IDE. Syntax highlighting was enabled, there were no code comments, and they never executed the code.

There were two kinds of programs. The first created two Rectangle objects, assigned the coordinates of the corners, and "drew" them on the screen. A printed question underneath the program asked the subjects whether the two rectangles overlapped (yes or no). The second program created four shape objects (choosing among Circles, Squares, Rectangles, and Triangles), and then "drew" them in some order on the screen. A printed multiple-choice question underneath the program asked subjects to tell us which three shapes were drawn on the screen last, and the order they were drawn in from five possible answers.

We used three instances of the first program in the experiment. The practice version of this program was used solely to familiarize the subject with the task. One experimental instance was identical to the practice version except for the use of local variables of non-mnemonic single letters. In contrast, the other experimental instance randomized and interleaved assignments of the corner coordinates for both rectangles. This program was designed to stress the subjects' abilities in spatial relations (deciding whether the two rectangles overlapped) and visual object grouping (interleaving the Rectangle initialization statements) and working memory (randomizing the order of the assignments prevents chunking each Rectangle's assignment sequence together, and fills up working memory to a greater amount).

There were seven instances of the second program. They differed in

(a) the order between initialization and drawing each shape (e.g. creating a shape and then drawing it, or creating all shapes and then drawing them in randomized order),

(b) the variable names (mnemonic vs generic) to impact subjects' working memory by interfering with their ability to remember the mapping between variable name and its shape,

(c) using an array to group the shapes and then looping over the array,

(d) making the loop construct mathematically more complex to stress the working memory (for remembering the order of shapes) and their mathematical skills,

(e) calling a separate function to swap the order,

(f) including a double-nested question-mark-colon conditional operator to engage the subject's mathematical and working memory abilities (comparing variables to constants).

Each of the tasks was designed to take subjects between 2 and 5 minutes to finish. Subjects could see both the code and the question on screen at the same time, and never needed to scroll. In fact, we directed the subjects to keep their hands still on the table to avoid affecting the EDA sensor through wrist motion.

3.4 Experimental Procedure

① When each subject first entered the lab, he was asked to fill out a consent form and a pre-questionnaire requesting demographic information. ② We synced the internal clock of the EDA sensor to the time on the eye-tracking computer and then placed it on the wrist of the subject's non-dominant hand (the hand that does not use the mouse). ③ We then connected the EDA sensor via Bluetooth to

the data recording computer and checked the live display to verify that a signal was being received. Since the EDA sensor works by detecting the electrical conductivity across the wrist, it sometimes fails to work if the subject has no sweat. For those few subjects who did not register any signal, we asked them to do a mild physical exertion (jumping jacks and walking up and down a flight of stairs) to cause them to sweat a little. This sufficed for all but three subjects, so we were unable to record their EDA signal.

During our pilot study, we had noticed that the subjects' EDA signal kept rising monotonically as they completed each subsequent task. This would cause an intense "learning" effect on the EDA signal, so we changed our protocol. (4) Prior to the first task, and in between each one, subjects were asked to watch two minutes of one of four different, calming, full screen YouTube videos of fish swimming in a fish tank and were requested to relax their minds. This relaxation caused the subjects' EDA measurements to return to baseline after about a minute. We were then able to use the EDA signal in the second minute of the video as a baseline for the EDA signal in the next task.

(5) Next we determined the dominant eye of the subject so we could be sure that our subsequent analysis of the eye-tracking gaze location data would point to the actual word that the subject was reading. (6) Each subject was asked to sit on non-wheeled chair in front of the eye-tracking computer and shift the seat around until their head stayed in an imaginary box about 50–75cm in front of the center of the screen. No chin rest or mouth guard was required. A peripheral display on a second computer enabled the experimenter to notice if the subject moved too far out of range ($>$ 37 cm side to side and/or $>$ 17 cm up and down) and ask him to move back into range before continuing.

The subjects were then shown the practice tasks in Visual Studio 2012, and the font size was adjusted, if requested by the subject. We asked them to think aloud while doing the task, and to tell us the answer out loud rather than typing it into the computer. We turned on the audio and video recording and helped them put the MindBand EEG sensor on their head. We then verified the MindBand's Bluetooth connection to the recording computer. Two of the subjects whose EDA signals were undetectable also exhibited problems with the MindBand, thus we were not able to record their MindBand either. We then calibrated the subjects' eye gaze using Tobii Studio's 9-point calibration program. We recalibrated any points that showed too much error. Finally, we began the experiment.

(7) The subjects were asked to watch the first fish tank video and (8) start their first task. (9) After each task, the subjects were given a paper-based NASA TLX survey instrument [32] to fill out that asked them to first rate the task from 1 – 20 along six dimensions: mental demand, physical demand, temporal demand, performance, effort, and frustration, and then compare each dimension with one another to determine the rank order of their importance. (10) Afterward, they watched the next two-minute fish tank video and continued to the next task. (11) Once the subjects finished their last task and NASA TLX survey, we removed all of the psycho-physiological sensors, and turned off the recordings. (12) Finally, we had them fill out a post-test questionnaire where they ranked the tasks they did according to the their own hind-sight perception of the tasks' difficulty. Subjects were able to go back and refamiliarize themselves with the task codes before ranking them.

3.5 Experimental Conditions

Every subject was expected to complete all ten tasks; first the two practice problems and then the eight experimental tasks. To combat any kind of learning effects caused by experience with the tasks, we counter-balanced the task order so that each subject took them in a different order. On average, it took the subjects 1:49 minutes (SD 1:37 minutes) to complete a task. The fastest subject completed one task of the second kind in 9 seconds. The slowest subject completed one task in 8:29 minutes, also of the second kind. Overall, each subject took about 1.5 hours to complete the entire experiment.

Some subjects failed to complete all the tasks before they had to leave. Two missed the final task, and one missed the last two tasks. Fortunately, we had no measurement difficulties with these three subjects.

4. DATA ANALYSIS

We collected psycho-physiological measurements for a total of 116 tasks. We present an overview of each sensor's measurements along with their related cognitive effects in Table 1. A detailed list of every measurement we used from the sensors is in Table 2.

For each subject's tasks, we also collected the completion time, the NASA TLX score, whether their answer was correct, and the difficulty rank they gave that task at the end of the study. We used the video recordings and the think-aloud protocols to fix any inadvertent mistakes we made during data analysis, which we describe next.

4.1 Data Cleaning and Transformation

Biometric data is notoriously noisy and contains large amounts of invalid data that must be cleaned before it can be analyzed.

Eye-Tracker. First, for each data point produced by the eye-tracker, an indication of the validity of the pupil size measurement enabled us to remove the invalid ones. Second, we noticed that the first pupil size measurement of each fixation occurring after a blink was suspiciously larger than the subsequent one (measured just 3.3 ms later). We learned that when your eyes close, even for a short time, the darkness causes your pupils to open just a little bit. To eliminate this artifact, we ended up eliding each of the first pupil size measurements after a blink.

Next, we compared the distribution of pupil sizes between subjects. We discovered that while each subject's pupil size distribution was Gaussian, the range of pupil sizes was very different. Consequently, to make subjects easier to compare, we standardized the pupil size measurement within participants by subtracting the mean from each value and dividing the difference by the standard deviation.

Pupil size tends to increase up to 0.5 mm under cognitive load, especially when reading difficult material. To find these events, we use a Matlab-based peak finding algorithm to count the number of peaks in the pupil size signal where the pupil size increased at least 0.1, 0.2, and 0.4 mm above its baseline. The baseline is calculated from the minimum and maximum pupil sizes gathered during each task as well as the prior one minute during the fish tank video.

Table 1: Overview of psycho-physiological measurements and the effects related to them in literature.

Measure	Previously found effect
Eyetracking	
Pupil size	Cognitive load [28, 40] Memory load [7]; Mental workload [36]
Saccades	Mental workload while air traffic control tasks [15]; Evaluation of user interfaces [24]
Fixations	Cognitive load while solving arithmetical tasks [35]; Performance during a code review [73, 65]; Effort to identify variable identifiers [65]
EEG	
Eye blinks	Visual attention [18]; Stress and anxiety level [20]; Classification of visual demanding tasks during flight [74]; Mental workload while air traffic control tasks [15]; Mental workload during arithmetic and visual tracking tasks [60]
Frequency bands (Alpha, Beta, Gamma, Delta, Theta)	Mental workload during air traffic control tasks [15]; Mental workload during arithmetic and visual tracking task [60]; Cognitive task classification [44]; Auditory awareness [46]
Ratios of frequency bands	Memory load during cognitive task [26]; Task engagement index [42, 43, 12]; Car driver status in various conditions [14]
Attention and Meditation	Cognitive load [27]
EDA	
Tonic	Anger and fear [3]; Mood states of bipolar patients [25]; Mental workload [74]; Arousal and engagement [48]
Phasic	Anger and fear [3]; Distinguish stress from cognitive load [62]; Arousal and engagement [48]

Table 2: Psycho-physiological measurements used from each of three sensors (abbreviated) (Δ represents the difference to the baseline).

Eyetracking (18)
NumSaccades/Min; SumSaccadeDuration/Min; {Mean, Median, Stdev}SaccadeDuration; NumFixations/Min; SumFixationDuration/Min; {Mean, Median, Stdev}FixationDuration; MinPupilSize; MaxPupilSize; Δ{Mean, Median, Stdev}PupilSize; ΔNumPupilSizeJumps>{0.1mm, 0.2mm, 0.4mm}
EEG (31)
{Min, Max}Attention; {Min, Max}Meditation; Δ{Mean, Stdev}Attention; Δ{Mean, Stdev}Meditation; ΔEyeblinks/Min; $\Delta(\alpha/\beta)$; $\Delta(\alpha/\gamma)$; $\Delta(\alpha/\delta)$; $\Delta(\alpha/\theta)$; $\Delta(\beta/\alpha)$; $\Delta(\beta/\gamma)$; $\Delta(\beta/\delta)$; $\Delta(\beta/\theta)$; $\Delta(\gamma/\alpha)$; $\Delta(\gamma/\beta)$; $\Delta(\gamma/\delta)$; $\Delta(\gamma/\theta)$; $\Delta(\delta/\alpha)$; $\Delta(\delta/\beta)$; $\Delta(\delta/\gamma)$; $\Delta(\delta/\theta)$; $\Delta(\theta/\alpha)$; $\Delta(\theta/\beta)$; $\Delta(\theta/\gamma)$; $\Delta(\theta/\delta)$; $\Delta(\theta/(\alpha+\beta))$; $\Delta(\beta/(\alpha+\theta))$
EDA (7)
{Min, Max}PeakAmpl; ΔNumPhasicPeaks/Min; ΔMeanPhasicPeakAmpl; ΔSumPhasicPeakAmpl/Min ΔMeanSCL; ΔAUCPhasic;

People's eyes move in small jerky movements called saccades, which each take under 75 ms. Someone can only read text when their eye fixates on a location between saccades. We extract the number and duration of a subject's eye saccades and fixations to gain insight into how their eye motion is impacted when reading material with various cognitive demands. Since every subject works at their own pace, we normalize many of our measurements by time to make them comparable between subjects.

EDA. EDA signals consist of two parts: a low frequency *tonic* signal which changes over a period of minutes, and a higher-frequency *phasic* signal, which takes 1–2 seconds to rise and 2–6 seconds to fall. The tonic component of the EDA signal, or skin conductance level (SCL), is commonly used as a measure of arousal. The phasic component reflects reactions based on external stimuli [61].

To clean our EDA signal, we first subtracted the signal's DC component to base it at 0 µS. We found a lot of noise in the signal from 2 Hz to 4 Hz, so we applied an exponential smoothing filter ($\alpha x(t)+(1-\alpha)x(t-1), \alpha = 0.08$). Next, we used a 5th order, low-pass Butterworth filter set to 0.05 Hz to extract the tonic signal. Since the maximum frequency of a phasic response is 0.33 Hz (the inverse of 6 seconds), we must extract the phasic signal at 0.66 Hz (the Nyquist sampling rate is twice the maximum frequency) to ensure we see the entire phasic response. Fortunately, the exponential smoothing we applied already eliminated the signal above 0.66 Hz, so we were able to use a high-pass version of the same Butterworth filter to extract the phasic signal.

The tonic SCL value must be measured relative to a recent baseline value. We calculate it by subtracting the mean SCL of the EDA signal while the subject watched the fish tank video from the one measured while the subject did each task.

The literature distinguishes between spontaneous changes in the EDA signal — called non-specific skin conductance responses (NS-SCRs) — and changes that occur after a specific stimuli — called event-related skin conductance responses (ER-SCRs) [2]. These changes are visible as peaks in the phasic signal which we found with a Matlab-based peak finder set to identify peaks with a minimum amplitude of 2 nS [22]. While NS-SCRs occur all the time, the only external stimuli the subjects could have experienced must have come from what they read during their program comprehension tasks. Thus, we can compute the likely number of ER-SCRs by subtracting the number of peaks experienced during the experimental task from the preceding one minute time period while they watched the fish tank video. We also use the peak finder to extract additional features from the signal, including the peak amplitude, frequency, and area under the curve (AUC) [3, 62], and normalize these by time.

EEG. The EEG sensor produces a raw signal sampled at 512 Hz. We first use a Matlab-based 60 Hz notch filter to remove signal noise caused by the overhead lights. To identify various mental states [11], we use Matlab's pwelch function

to compute the power spectrum distribution for each of the five familiar brain wave frequency bands: Alpha (α) (8–12 Hz), Beta (β) (12–30 Hz), Gamma (γ) (30–80 Hz), Delta (δ) (0–4 Hz) and Theta (θ) (4–8 Hz) [31]. Since every person has a unique power spectrum distribution, we compute the ratio of each band with one another in order to compare the values between individuals. In addition, inspired by Kramer and Lee [42, 44], we compute $Beta(\beta)/(Alpha(\alpha) + Theta(\theta))$ and $Theta(\theta)/(Alpha(\alpha) + Beta(\beta))$ as additional measures of task difficulty.

We found an additional use for the EEG sensor. Due to its placement on the forehead, the sensor is exquisitely sensitive to the motor signals of the face, such as brow furrowing, eyebrow motion, and blinking. Each of these motor activities produces a high amplitude, low frequency signal which is easy to distinguish from neuronal activity. Brookings *et al.* showed that a person's blink rate decreases significantly when tasks become more difficult [15]. Taking advantage of a technique illustrated by Manoilov [47], we use a band-pass Butterworth filter to filter our EEG signal from 0.5 Hz to 3 Hz and apply our Matlab-based peak finding algorithm to find peaks that are over 100x stronger than the waveform's average amplitude. These peaks correspond to eye blinks. We calculate the number of blinks per minute and then subtract out the baseline number of blinks during the subject's prior viewing of the fish tank video.

Finally, we extract the pre-computed 1 Hz Attention and Meditation signals from the NeuroSky EEG sensor, and compute the mean, median, standard deviation, minimum, and maximum values for our analysis.

4.2 Outcome Measures

We used two outcome measures in our tasks: the NASA Task Load Index (TLX) [32] filled out on paper after each task, and a subjective ranking of tasks based on the subject's *post hoc* perception of their difficulty at the end of the experimental session. The NASA TLX is a commonly-used subjective measure for assessing cognitive load [27]. After each task, the subject rates its difficulty on six 20-point scales: performance (good/poor), mental demand (low/high), physical demand (low/high), temporal demand (low/high), effort (low/high), and frustration (low/high). Each scale is defined for the subject using Hart and Staveland's instructions [50] along with a discussion of their meaning with the experiment administrator. After marking the six ratings, the subject then considers every possible pair of scale names, and is instructed to circle the scale name in each pair which is more important to his experience of workload than the other. We compute the overall NASA TLX score to be the sum of the products of each rating and the tally (0–5) of the number of times it was chosen as more important, and then divided by 15. Three of the authors computed these scores at different times during analysis to ensure we transcribed and calculated them properly.

While the NASA TLX score gives us insight into the subject's mental workload for each question, we were interested in a measure of the subject's summative assessment of task difficulty. To this end, we asked each subject to rank the ten tasks he did from easy to hard (ties were acceptable). A few subjects wrote down additional comments to clarify how they thought about the difficulty (e.g. "The Rectangle tasks were difficult because I am terrible at doing spatial relations in my head.").

To make prediction simpler for our machine learning algorithm, we nominalized the task difficulty ranking as easy or difficult. Low ranks were changed to easy and high ranks were labeled difficult. For scores in the middle, we looked at each subject's additional comments and found that in all but two cases out of 116, subjects clearly expressed where there was an easy/difficult gap in their perception of the tasks' difficulty. For the other two cases, we were able to use the NASA TLX score to disambiguate (in favor of correlation) because there the NASA TLX score was clearly unambiguous. After nominalizing the task difficulty ranking, our dataset consisted of 51 difficult and 65 easy tasks.

To validate the task difficulty ranking, we confirmed that there was a high correlation between the task difficulty ranking and the NASA TLX scores. A Spearman correlation shows that the NASA TLX score is correlated with the subjects' task difficulty rankings ($r[116] = 0.587, p < 0.01$). The NASA TLX easy/difficult boolean was also similar to the Task Difficulty easy/difficult boolean ($\chi^2(1, 116) = 57.954$, $p < 0.01$) with an accuracy of 85%.

As a final step in validating the task difficulty ranking, we looked at the correlation between the ranking and task completion time, since time on task is also a proxy for difficulty [27]. A Spearman correlation of $r[116] = 0.724$ ($p < 0.01$) supports this correlation, and thus our choice of task difficulty ranking for our outcome measure.

4.3 Machine Learning

Machine learning has been shown to be a promising approach to find links between low-level data capture and high-level phenomena of interest [10]. We used Weka [30], a popular, Java-based machine learning classification toolkit, to develop a set of classifiers that can connect our psycho-physiological measures with task difficulty.

There were a number of parameters that could affect the design of the classifier we wished to develop. First, and foremost, was a choice between three types of predictions: by participant, by task, and by participant-task pair. The by-participant classifier would be the most useful in practice — trained on a small set of people doing program comprehension tasks, it could be applied to any new person doing new tasks and still accurately assess task difficulty. Next, in utility, is the by-task classifier; when trained on people doing a set of tasks, it would work well when applied to one of those people doing any new task. Finally, the by-participant-task pair classifier shows the proof-of-concept — trained on a set of people doing programming tasks, it can predict the difficulty of the task as perceived by one of those people doing a task that the rest already did. These three predictions were used to stratify the datasets into test and training sets.

Second, was the choice of classification algorithm. We considered Naive Bayes, a J48 decision tree (using Weka's implementation of C4.5 [58]), and a Support Vector Model. For our goal of an instantaneous classifier, Naive Bayes was the best choice because of the ease in which its training can be updated on-the-fly, improving its performance as it adjusts to its user.

Third, we can train the classifier on the entire set of data from each participant and task, or divide up the data collection into sliding time windows. This would enable us to create a classifier usable before a developer finished his task that would adjust to his changing psycho-physiological conditions. We divided up our data using sliding time win-

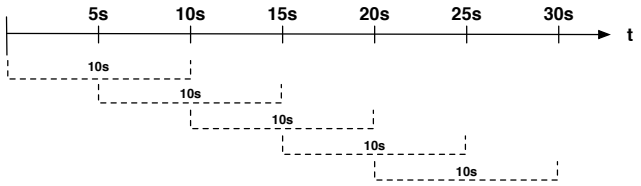

Figure 1: Sliding windows of size 10 seconds with 5 second offsets.

dows of sizes from 5 seconds to 60 seconds, sliding 5 seconds between intervals. A demonstration of this is shown in Figure 1.

Finally, the last parameter to the machine learning algorithm is to identify the best set of measurements (features) that will be used to train the model [21, 26]. We chose to experiment with measurements extracted from every combination of our three sensors (7 possible sets of features). To ensure correct performance for Naive Bayes, we removed five measurements that correlated almost perfectly with measurements we left in.

5. RESULTS

This section reports the results of our use of machine learning to define classifiers to predict task difficulty.

5.1 Task Difficulty Classification

To evaluate whether we can use psycho-physiological measures to predict if a task is easy or difficult (RQ1), we perform a post-hoc analysis that applies machine learning to the data gathered over the whole task period. We used a leave-one-out strategy to create an exhaustive set of test and training folds to train classifiers using all of the sensors for each stratification (by participant, by task, by participant-task). The average precision, recall, and f-measure for the three classifiers we trained is shown on the last row of each section in Table 3. The best overall performance comes when predicting a new task with 84.38% precision and 69.79% recall.

5.2 Evaluating Sensors

Next, we wished to find out how well each of the three sensors, eye-tracking, EDA, and EEG, could be used to predict task difficulty (RQ2). We trained classifiers on each combination of sensors creating training and test sets for all three predictions (by participant, by task, by participant-task) over the entire dataset. The results are shown in Table 3. Considering each sensor by itself, the eye-tracker has the best predictive power for new participants (65.10% f-measure). When predicting a new task, EEG has the highest precision (81.97%), but the eye-tracker has the best recall (66.67%). When predicting a participant-task pair, the eye-tracker comes out on top (66.67% f-measure). Combinations of sensors performed better when predicting new tasks (all sensors) and participant-task pairs (eye-tracker+EDA).

We investigated whether eliminating features from each sensor could help improve the accuracy of our classifiers. We used Weka's CfsSubsetEval algorithm [58] to analyze the features for each sensor combination and keep those that correlated highly with the outcome variable and poorly with other features. In some cases, this yielded better performance on our data (e.g. EDA+EEG f-measure rose from 62.01% to 69.73%), however running ANOVA tests on the Weka output failed to show any significant differences between the original and shrunken sets of features. Thus, the improvement we saw may be an artifact of our dataset; capturing additional input data would help establish whether feature elimination will truly improve performance.

Table 3: Performance characteristics of classifiers trained on the entire dataset over data from all possible combinations of three sensors to predict a participant, a task, and a participant-task pair. The best measurements for a prediction are bold.

Prediction	Sensors	Precision	Recall	F-Measure
By Participant	**Eye**	**69.16%**	**65.83%**	**65.10%**
	EDA	55.18%	55.77%	51.99%
	EEG	53.05%	56.73%	50.82%
	Eye+EDA	68.37%	64.42%	61.92%
	Eye+EEG	68.58%	63.46%	60.89%
	EDA+EEG	68.02%	64.58%	62.01%
	Eye+EDA+EEG	64.99%	64.58%	62.21%
By Task	Eye	79.17%	66.67%	69.65%
	EDA	75.12%	58.65%	63.80%
	EEG	81.97%	59.62%	63.40%
	Eye+EDA	78.59%	66.35%	70.37%
	Eye+EEG	82.42%	66.35%	69.89%
	EDA+EEG	82.79%	65.63%	69.76%
	Eye+EDA+EEG	**84.38%**	**69.79%**	**73.33%**
By Participant-Task	Eye	66.67%	66.67%	66.67%
	EDA	59.62%	59.62%	59.62%
	EEG	56.73%	56.73%	56.73%
	Eye+EDA	**68.27%**	**68.27%**	**68.27%**
	Eye+EEG	62.50%	62.50%	62.50%
	EDA+EEG	62.50%	62.50%	62.50%
	Eye+EDA+EEG	67.71%	67.71%	67.71%

5.3 Evaluating Time Windows

Finally, to see if we could build a classifier that would be accurate if receiving streaming data from the sensors as the developer worked, we built a set of classifiers trained on sliding time windows (RQ3). However, we needed to find out which time window sizes would work the best. In some cases, the window size was longer than the task data, so we just included the time windows that were available. Figure 2 presents the precision for each classifier trained on a particular time window size using all of the available sensors. There appears to be no major differences in the performance of that classifier over the various time windows.

However, we calculated the effects of combining a subset of sensors along with the use of sliding time windows. We found the best classifier for predicting new participants to use just the eye-tracker and the EDA sensor with a time window size of 60 seconds. This performed at 70.46% precision and 62.20% recall, which is just a tiny bit better than using all three sensors or just the eye-tracker. For predicting tasks, the best classifier used just the EDA sensor with a 20 second time window and got a precision of 83.74% and a recall of 64.12%. This performs better than using all three sensors on sliding time window data, but not better than when trained on the entire dataset. When predicting a participant-task, the best classifier used the EDA and EEG sensors with a time window of 55 seconds. This achieved a precision of 100.00% and a recall of 66.13%, which is better than both using all of the sensors and the entire dataset.

6. DISCUSSION

The results of our machine learning experiments answer RQ1, demonstrating that it is possible to very accurately predict whether a task is easy or difficult using psycho-

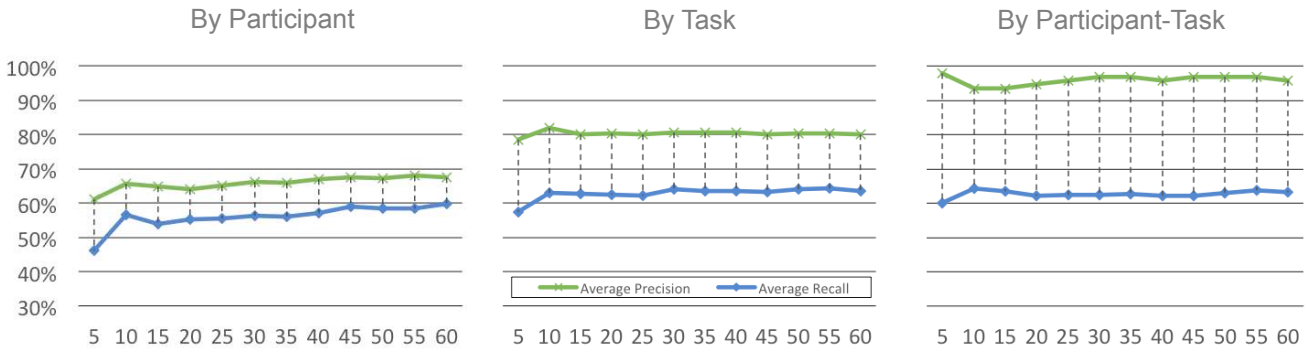

Figure 2: Precision and recall using all sensors over time windows of 5–60 seconds.

physiological measures. Using all of the task data, a classifier trained on the three sensors achieves 64.99% precision and 64.58% recall when predicting our nominalized task difficulty measure for new participants. The performance increases to 84.38% precision and 69.79% recall when predicting new tasks, likely because the classifier has had a chance to see the participant in action on other tasks. For predicting participant-task pairs, the precision and recall both settle at 67.71%.

Answering RQ2, when we checked which combinations of sensors had the best predictive power, we found that for predicting a new participant, the eye tracker did the best (69.16% precision and 65.83% recall); when predicting a new task, the combination of all three sensors was best (84.38% precision and 69.79% recall); and if predicting a participant-task pair, the pair of eye tracking and EDA sensors was best (68.27% precision and 68.27% recall). Thus, for predicting new participants and new participant-task pairs, it may be better to use a subset of the sensors available to achieve better performance (and save money!).

When we measured the predictive power of classifiers that use sliding time windows, we found that for predicting new participants, the best classifier using all three sensors uses a time window of 55 seconds, and reaches 68.04% precision and 58.55% recall. The best classifier for predicting a new task uses a 30 second time window and reaches a precision of 80.68% and a recall of 64.01%. Finally, for predicting a participant-task pair, the best classifier uses a time window of 55 seconds, and achieves a precision of 96.74% and a recall of 63.73%. Compared with using the entire dataset, dividing the data into sliding time windows is beneficial for predicting new participants and participant-tasks, but not for predicting new tasks.

When we combined the use of sliding time windows with subsets of sensors, we found it possible to improve the precision and recall slightly, compared with using the entire dataset and all of the sensors. The big improvement came for predicting a participant-task pair with just the EDA and EEG sensors (our two lowest cost sensors) and a time window of 55 seconds.

Our work provides an existence proof that answers RQ3. It is possible to use low-cost, off-the-shelf psycho-physiological sensors to develop accurate classifiers. The existence of such an indicator should provide many opportunities for new software engineering tools. For instance, it could be used to detect the places in the code that developers have difficulties with while working, and mark them for review or for future refactoring. As pointed out by Bailey and Iqbal, it could also help prevent interruptions during particularly difficult tasks which might require a longer task resumption time [4].

7. THREATS TO VALIDITY

We describe several threats to the validity of our study in this section.

External Validity. While we feel these results should be generalizable to other kinds of short, code comprehension tasks, more work remains to be done to validate our classifier against tasks that are longer, contain more code, and involve code creation and maintenance. We mitigated this risk somewhat by carefully constructing the tasks to vary their difficulty and effects on various brain functions according to past empirical results on programmers [19]. We do not claim that these results are generalizable to students, novice software developers, or other broader populations, but feel that our use of professional developers situates this work within a population that creates most of the software in the world.

Internal Validity. During the study, the participants were required to complete a series of small tasks of varying difficulty. We counter-balanced the task order to combat learning effects, but did not have a large enough population to explore every possible order. Thus, some learning effects may have gone unnoticed. Our study took place in a lab setting, thus, our subjects may have performed differently than in their own work environments. Since the typical effect of lab studies on subjects is to *increase* their performance, due to their desire to please the experimenter, they may have experienced less task difficulty than normal.

While the tasks in our study were not very long (only several minutes), we believe the subjects' behavioral responses (interpreted with their think-aloud narrative) to be fairly typical of software developers working on their own longer-term tasks. As noted in Section 3.3, even short, lab-based, experimental tasks like ours can be designed to provoke cognitive difficulties in many of the different functional brain regions that comprise software development skills. More experiments will be required to establish whether the trends we have seen in our data apply to different programming tasks. While there is great individual variability in the performance of software developers (which we also observed in our experimental subjects), there apparently was not enough in the sensor data to impede accurate classification. A study

with a larger and more diverse sample of developers should be able to tell us whether the classifiers will be confounded, or confirm their generalizability. Even if the classifiers fail to generalize due to individual participant variation, our use of the Naive Bayes algorithm will support performance improvement through additional classifier training by the participant while he works. Given the great number of hours that developers spend in front of their computers and the potential utility of such an instantaneous classifier of task difficulty, training classifiers for each individual programmer to improve accuracy should be a palatable tradeoff.

Construct Validity. The goal of this study was to investigate the predictive power of multiple psycho-physiological measures for task difficulty. A threat to the study is that there are other factors that might either influence the perceived task difficulty or psycho-physiological measurements unrelated to task difficulty itself. These include personality traits, private and professional stress, or even the time of day. We tried to mitigate the risk by providing the same quiet environment for every subject, but we may need to investigate these effects in the future. Second, to make predictions simple for our machine learning approach, we categorized the task difficulty ranking into easy or difficult. This binary classification might have lead to better results than if we had used the original interval scale. Third, our tasks were constructed to be varying shades of difficult, but this is really subjective. We triangulated this difficulty using not just a retrospective ranking of tasks by difficulty, but also through the commonly used NASA TLX score. These measures correlated together quite well, especially when both were converted to nominal form. Thus, we do believe that our task difficulty construct is quite valid.

8. CONCLUSION

Software developers regularly experience difficulties in their work that waste their time and may cause them to introduce bugs into their software. Previous research focused on identifying bug risk using correlations between defects and various software process metrics. Our research, however, is the first to investigate an automated approach using psycho-physiological sensor data to detect both *post hoc* and as the developer works, whether a developer perceives that his program comprehension task is difficult. Our experimental results show that we can train a Naive Bayes classifier on short or long time windows with a variety of sensor data to predict whether a new participant will perceive his tasks to be difficult with a precision of over 70% and a recall over 62%. Our results also demonstrate that it is possible to use fewer sensors and still retain the ability to accurately classify task difficulty. Now that we have shown that these classifiers can be built, researchers can leverage them to develop novel programming support tools, allowing them to potentially intervene in time to stop bugs from entering the code.

9. REFERENCES

[1] N. Ali, Z. Sharafl, Y. Gueheneuc, and G. Antoniol. An empirical study on requirements traceability using eye-tracking. In *Proc. of the 28th Int'l. Conf. on Software Maintenance (ICSM)*, pages 191–200, 2012.

[2] J. L. Andreassi. *Psychophysiology: Human Behavior & Physiological Response*. Lawrence Erlbaum Associates, 2007.

[3] A. F. Ax. The physiological differentiation between fear and anger in humans. *Psychosomatic Medicine*, pages 433–442, 1953.

[4] B. P. Bailey and S. T. Iqbal. Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Trans. on Computer-Human Interaction*, 14(4):21, 2008.

[5] L. O. Bauer, B. D. Strock, R. Goldstein, J. A. Stern, and L. C. Walrath. Auditory discrimination and the eyeblink. *Psychophysiology*, 22(6):636–641, 1985.

[6] J. Beatty. Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological Bulletin*, 91(2):276, 1982.

[7] J. Beatty. Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological Bulletin*, 91(2):276–292, March 1982.

[8] R. Bednarik and M. Tukiainen. An eye-tracking methodology for characterizing program comprehension processes. In *Proc. of the 2006 Symposium on Eye Tracking Research & Applications*, pages 125–132. ACM, 2006.

[9] R. Bednarik and M. Tukiainen. Temporal eye-tracking data: evolution of debugging strategies with multiple representations. In *Proc. of the 2008 Symposium on Eye Tracking Research & Applications*, pages 99–102. ACM, 2008.

[10] R. Bednarik, H. Vrzakova, and M. Hradis. What do you want to do next: a novel approach for intent prediction in gaze-based interaction. In *Proc. of the Symposium on Eye Tracking Research & Applications*, pages 83–90. ACM, 2012.

[11] H. Berger. Über das Elektrenkephalogramm des Menschen. In *European Archives of Psychiatry and Clinical Neuroscience*, volume 87, pages 527–570, 1929.

[12] C. Berka, D. J. Levendowski, M. N. Lumicao, A. Yau, G. Davis, V. T. Zivkovic, R. E. Olmstead, P. D. Tremoulet, and P. L. Craven. EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks. *Aviation, space, and environmental medicine*, pages B231–B244, 2007.

[13] W. Boucsein. *Electrodermal Activity*. Springer-Verlag, New York, 2012.

[14] K. A. Brookhuis and D. De Waard. The use of psychophysiology to assess driver status. In *Ergonomics*, 1993.

[15] J. B. Brookings, G. F. Wilson, and C. R. Swain. Psychophysiological responses to changes in workload during simulated air traffic control. *Biological psychology*, 42(3):361–377, 1996.

[16] M. Crosby and J. Stelovsky. How do we read algorithms? A case study. *Computer*, 23(1):25–35, 1990.

[17] M. E. Dawson, A. M. Schell, and D. L. Filion. The electrodermal system. *Handbook of Psychophysiology*, page 159, 2007.

[18] P. J. De Jong and H. Merckelbach. Eyeblink frequency, rehearsal activity, and sympathetic arousal. *Int'l. Journal of Neuroscience*, 51(1-2):89–94, 1990.

[19] F. Détienne. *Software Design - Cognitive Aspects*. Springer-Verlag New York, Inc., New York, NY, USA,

2002.

[20] D. G. Doehring. The relation between manifest anxiety and rate of eyeblink in a stress situation. Technical report, Central Institute for the Deaf, St Louis, MO, 1957.

[21] S. Eivazi and R. Bednarik. Predicting problem-solving behavior and performance levels from visual attention data. In *Proc. of the 2nd Workshop on Eye Gaze in Intelligent Human Machine Interaction*, pages 9–16, 2011.

[22] D. C. Fowles, M. J. Christie, R. Edelberg, W. W. Grings, D. T. Lykken, and P. H. Venables. Publication recommendations for electrodermal measurements. *Psychophysiology*, 18(3):232–239, 1981.

[23] A. Gevins, M. E. Smith, H. Leong, L. McEvoy, S. Whitfield, R. Du, and G. Rush. Monitoring working memory load during computer-based tasks with EEG pattern recognition methods. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 40(1):79–91, 1998.

[24] J. H. Goldberg and X. P. Kotval. Computer interface evaluation using eye movements: methods and constructs. *Int'l. Journal of Industrial Ergonomics*, 24(6):631 – 645, 1999.

[25] A. Greco, A. Lanata, G. Valenza, G. Rota, N. Vanello, and E. Scilingo. On the deconvolution analysis of electrodermal activity in bipolar patients. In *Proc. of the Annual Int'l. Conf. of the IEEE on Engineering in Medicine and Biology Society (EMBC)*, pages 6691–6694, 2012.

[26] D. Grimes, D. S. Tan, S. E. Hudson, P. Shenoy, and R. P. Rao. Feasibility and pragmatics of classifying working memory load with an electroencephalograph. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 835–844. ACM, 2008.

[27] E. Haapalainen, S. Kim, J. F. Forlizzi, and A. K. Dey. Psycho-physiological measures for assessing cognitive load. In *Proc. of the 12th ACM Int'l. Conf. on Ubiquitous Computing*, pages 301–310. ACM, 2010.

[28] E. Haapalainen, S. Kim, J. F. Forlizzi, and A. K. Dey. Psycho-physiological measures for assessing cognitive load. In *Proc. of the 12th ACM Int'l. Conf. on Ubiquitous Computing*, Ubicomp '10, pages 301–310, New York, NY, USA, 2010. ACM.

[29] A. Haarmann, W. Boucsein, and F. Schaefer. Combining electrodermal responses and cardiovascular measures for probing adaptive automation during simulated flight. *Applied Ergonomics*, 40(6):1026–1040, 2009.

[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, Nov. 2009.

[31] T. C. Handy. *Event-related potentials: a methods handbook*. MIT Press, 2005.

[32] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human Mental Workload*, 1(3):139–183, 1988.

[33] M. K. Holland and G. Tarlow. Blinking and mental load. *Psychological Reports*, 31(1):119–127, 1972.

[34] M. K. Holland and G. Tarlow. Blinking and thinking. *Perceptual and Motor Skills*, 41(2):403–406, 1975.

[35] C. S. Ikehara and M. E. Crosby. Assessing cognitive load with physiological sensors. In *Proc. of the 38th Annual Hawaii Int'l. Conf. on System Sciences*, HICSS '05, pages 295.1–, Washington, DC, USA, 2005. IEEE Computer Society.

[36] S. T. Iqbal, X. S. Zheng, and B. P. Bailey. Task-evoked pupillary response to mental workload in human-computer interaction. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1477–1480, New York, NY, USA, 2004. ACM.

[37] R. J. Jacob and K. S. Karn. Commentary on section 4 - eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The Mind's Eye*, pages 573 – 605. North-Holland, Amsterdam, 2003.

[38] I. A. Khan, W.-P. Brinkman, and R. M. Hierons. Do moods affect programmers' debug performance? *Cognition, Technology & Work*, 13(4):245–258, 2011.

[39] I. A. Khan, R. M. Hierons, and W. P. Brinkman. Mood independent programming. In *Proc. of the 14th European Conf. on Cognitive ergonomics: Invent! Explore!*, pages 269–272. ACM, 2007.

[40] J. Klingner. Fixation-aligned pupillary response averaging. In *Proc. of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pages 275–282, New York, NY, USA, 2010. ACM.

[41] A. F. Kramer. Physiological metrics of mental workload: A review of recent progress. Technical Report NPRDC-TN-90-23, Navy Personnel Research and Development Center, June 1990.

[42] A. F. Kramer. Physiological metrics of mental workload: A review of recent progress. *Multiple-task Performance*, pages 279–328, 1991.

[43] I. Lawrence J. Prinzel, P. Alan T., F. Frederick G., S. Mark W., and M. Peter J. Empirical analysis of EEG and ERPs for pyschophysiological adaptive task allocation. Technical report, NASA Langley, 2001.

[44] J. C. Lee and D. S. Tan. Using a low-cost electroencephalograph for task classification in HCI research. In *Proc. of the 19th Annual ACM Symposium on User Interface Software and Technology*, pages 81–90. ACM, 2006.

[45] C. Maaoui and A. Pruski. Emotion recognition through physiological signals for human-machine communication. *Cutting Edge Robotics*, 2010.

[46] S. Makeig and T.-P. Jung. Tonic, phasic, and transient EEG correlates of auditory awareness in drowsiness. *Cognitive Brain Research*, 4(1):15 – 25, 1996.

[47] P. Manoilov. Eye-blinking artefacts analysis. In *Proc. of the 2007 Int'l. Conf. on Computer Systems and Technologies*, page 52, 2007.

[48] D. McDuff, A. Karlson, A. Kapoor, A. Roseway, and M. Czerwinski. Affectaura: an intelligent system for emotional memory. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 849–858. ACM, 2012.

[49] A. Nakasone, H. Prendinger, and M. Ishizuka. Emotion recognition from electromyography and skin conductance. In *Proc. of the 5th Int'l. Workshop on Biosignal Interpretation*, pages 219–222, 2005.

[50] NASA - Ames Research Center, Aerospace Human Factors Research Division. NASA TLX Load Index (TLX): Paper-and-pencil version.

[51] Neurosky. Neurosky's eSense™ meters and detection of mental state, 2009.

[52] S. Nevalainen and J. Sajaniemi. Short-term effects of graphical versus textual visualisation of variables on program perception. In *Proc. of the 17th Annual Psychology of Prgramming Interest Group Workshop*, pages 77–91, 2005.

[53] N. Nourbakhsh, Y. Wang, F. Chen, and R. A. Calvo. Using galvanic skin response for cognitive load measurement in arithmetic and reading tasks. In *Proc. of the 24th Australian Computer-Human Interaction Conf.*, OzCHI '12, pages 420–423, New York, NY, USA, 2012. ACM.

[54] C. Parnin. Subvocalization-toward hearing the inner thoughts of developers. In *Proc. of the 19th Int'l. Conf. on Program Comprehension (ICPC)*, pages 197–200. IEEE, 2011.

[55] R. W. Picard, E. Vyzas, and J. Healey. Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(10):1175–1191, 2001.

[56] M.-Z. Poh, N. Swenson, and R. Picard. A wearable sensor for unobtrusive, long-term assessment of electrodermal activity. *IEEE Trans. on Biomedical Engineering*, 57(5):1243–1252, 2010.

[57] G. C. Porras and Y.-G. Guéhéneuc. An empirical study on the efficiency of different design pattern representations in UML class diagrams. *Empirical Software Engineering*, 15(5):493–522, 2010.

[58] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[59] D. W. Rowe, J. Sibert, and D. Irwin. Heart rate variability: Indicator of user state as an aid to human-computer interaction. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 480–487. ACM Press/Addison-Wesley Publishing Co., 1998.

[60] K. Ryu and R. Myung. Evaluation of mental workload with a combined measure based on physiological indices during a dual task of tracking and mental arithmetic. *Int'l. Journal of Industrial Ergonomics*, 35(11):991–1009, 2005.

[61] S. Schmidt and H. Walach. Electrodermal activity (EDA) - state-of-the-art measurements and techniques for parapsychological purposes. *Journal of Parapsychology*, 64(2):139, June 2000.

[62] C. Setz, B. Arnrich, J. Schumm, R. La Marca, G. Troster, and U. Ehlert. Discriminating stress from cognitive load using a wearable EDA device. *IEEE Trans. on Information Technology in Biomedicine*, 14(2):410–417, 2010.

[63] Z. Sharafi, Z. Soh, Y.-G. Guéhéneuc, and G. Antoniol. Women and men - different but equal: On the impact of identifier style on source code reading. In *Proc. of the 20th Internationl Conf. on Program Comprehension*, pages 27–36. IEEE, 2012.

[64] Z. Sharafl, A. Marchetto, A. Susi, and G. Antoniol. An empirical study on the efficiency of graphical vs. textual representations in requirements comprehension. In *Proc. of the Int'l. Conf. on Program Comprehension*. IEEE, 2013.

[65] B. Sharif, M. Falcone, and J. I. Maletic. An eye-tracking study on the role of scan time in finding source code defects. In *Proc. of the Symposium on Eye Tracking Research & Applications*, pages 381–384. ACM, 2012.

[66] B. Sharif and J. I. Maletic. An eye tracking study on camelcase and under_score identifier styles. In *Proc. of the 18th Int'l. Conf. on Program Comprehension (ICPC)*, pages 196–205. IEEE, 2010.

[67] B. Sharif and J. I. Maletic. An eye tracking study on the effects of layout in understanding the role of design patterns. In *Proc. of the Int'l. Conf. on Software Maintenance*, pages 1–10. IEEE, 2010.

[68] Y. Shi, N. Ruiz, R. Taib, E. Choi, and F. Chen. Galvanic skin response (GSR) as an index of cognitive load. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*, pages 2651–2656. ACM, 2007.

[69] J. Simola, J. Salojärvi, and I. Kojo. Using hidden markov model to uncover processing states from eye movements in information search tasks. *Cognitive Systems Research*, 9(4):237–251, 2008.

[70] M. E. Smith and A. Gevins. Neurophysiologic monitoring of mental workload and fatigue during operation of a flight simulator. In *Defense and Security*, pages 116–126. Int'l. Society for Optics and Photonics, 2005.

[71] M. Sterman, C. Mann, and D. Kaiser. Quantitative EEG patterns of differential in-flight workload. In *Proc. of the 6th Annual Workshop on Space Operations Applications and Research (SOAR)*, volume 2, NASA, Johnson Space Center, 1993.

[72] C. Telford and N. Thompson. Some factors influencing voluntary and reflex eyelid responses. *Journal of Experimental Psychology*, 16(4):524, 1933.

[73] H. Uwano, M. Nakamura, A. Monden, and K.-I. Matsumoto. Analyzing individual performance of source code review using reviewers' eye movement. In *Proc. of the 2006 Symposium on Eye Tracking Research & Applications*, ETRA '06, pages 133–140, New York, NY, USA, 2006. ACM.

[74] G. F. Wilson. An analysis of mental workload in pilots during flight using multiple psychophysiological measures. *The Int'l. Journal of Aviation Psychology*, 12(1):3–18, 2002.